

REMARKS

Claims 38-55 have been amended. Claims 1-80 remain pending in the application. Claims 1-17 and 56-80 have been withdrawn. Reconsideration is respectfully requested in light of the following remarks.

Double Patenting Rejection:

The Examiner provisionally rejected claims 18-28, 33, 35, 37-46, 51, 53 and 55 under the judiciary created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 3, 8-10 and 13-17 of co-pending Application No. 10/692,913. Should this rejection become non-provisional, Applicant will consider filing a terminal disclaimer or present reasons traversing the rejection.

Section 101 Rejection:

The Examiner rejected claims 38-55 under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Applicant respectfully traverses this rejection; however, to expedite prosecution, Applicant has amended claims 38-55 to recite *a computer-accessible storage medium storing program instructions, wherein the program instructions are computer-executable to implement...* . Applicant respectfully requests removal of the § 101 rejection of claims 38-55.

Section 102(e) Rejection:

The Examiner rejected claims 18-55 under 35 U.S.C. § 102(e) as being anticipated by Conallen (“Building Web Applications with UML: Second Edition, October 10, 2002”). Applicant respectfully traverses this rejection for at least the following reasons.

As a first matter, Conallen is generally **not directed at Web Services**, but is **instead generally directed at Web applications**. On page 3, first paragraph, Conallen states (emphasis added), “this book is about building model-driven Web applications.” The term Web Services is well known in the art, and one of ordinary skill in the art would recognize the difference between the terms Web Services and Web applications. The background section of the instant application provides an extensive discussion of Web Services. Furthermore, Conallen defines Web applications thusly in the paragraph beginning on page 9 and extending onto page 10 (emphasis added):

In its simplest terms, a Web application is a Web system that allows its users to execute business logic with a Web browser...There is a subtle difference between a Web application and a Web site. For the purpose of this book, a Web application is a Web site where user input – navigation through the site and data entry – affects the state of the business: beyond, of course, access logs and hit counters. In essence, a Web application uses a Web site as the front end to a business application.

Conallen does briefly discuss Web Services on pp. 63-68, in a section titled “Web Services” that appears in Chapter 4, titled “Beyond HTTP and HTML,” which begins on page 49. Conallen makes clear the distinction between Web Services and Web applications, for example in the third paragraph on page 63:

The term Web Services is the latest hot phrase in development circles. Although the term has the word Web in it, **it is not a Web application-specific technology**. Instead, it uses Web technologies, such as Web servers and HTTP, to provide a set of services that can be invoked by other programs on the network. (emphasis added)

On page 49, second paragraph, Conallen discusses the content and reason for Chapter 4:

With the recent successes of Web applications, more and more architects are choosing this architecture for their next generation of systems. The significant advantages of easy deployment and minimal client configuration are well suited to organizations that maintain a varied array of computer types and models. This increased use of the Web as an architectural platform, however, has stretched the limits of the ability for HTTP and HTML to deliver the functionality required in relatively sophisticated software systems. This chapter discusses the limitations and extensions to these two principal elements of Web applications: HTTP and HTML.

Thus, both Conallen and Applicant's specification are consistent in distinguishing Web Services from Web applications. The term "Web Services" is a well-understood term of art. Most of the teachings of Conallen cited by the Examiner pertain to Web applications, not Web Services as recited in Applicant's claim. **Nowhere does Conallen extend the notions presented in his book to generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components.** Again, Conallen clearly states "this book is about building model-driven Web applications."

In regard to claim 18, contrary to the Examiner's assertion, Conallen does not anticipate a system for generating a vendor-independent Web Service architecture. As noted above, Conallen is generally not directed at Web Services, but is instead generally directed at Web applications. Furthermore, nowhere does Conallen extend the notions presented in his book to generating a vendor-independent Web Service architecture. The Examiner cites Conallen, page 65, and states "using UDDI, a standard for publishing and describing Web services" in support of the assertion that Conallen anticipates a system for generating a vendor-independent Web Service architecture. UDDI is **not** a system for generating a vendor-independent Web service architecture; UDDI is simply one component or tool that may be used in or with Web service architectures. As Conallen points out on page 65, in the section cited by the Examiner, UDDI is instead "a mechanism for publishing and describing Web services to potential clients." On page 65, Conallen further states "A UDDI Business Registry is a set of replicated registries of information about Web services on a network." On page 66, Conallen states that the "general usage scenario [for UDDI] is for a programmer to use a Web-based interface or specialized tool to query the UDDI registry via its inquiry API." UDDI is clearly not used to generate a vendor-independent Web Service architecture; UDDI is used to register and publish descriptive information about Web services on a network, and to query for the published descriptive information.

To further clarify that UDDI is not used to generate a vendor-independent Web Service Architecture, Conallen goes on to state, on page 66, "This information described

in a UDDI business service...categorizes and points to URLs that describe Web services but doesn't provide enough detail for a programmer to code a system that can accept and send SOAP-based Web service messages."

In further regard to claim 18, contrary to the Examiner's assertion, Conallen does not anticipate a system for generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components. The Examiner cites page 425, "web server is most likely a commodity component, such as Tomcat, WebSphere, WebLogic and IIS." Page 425 appears in Appendix D, titled "Master Template Pattern," which begins on page 423, and presents a "Logical View." The first paragraph on page 423, titled "Overview," states (emphasis added):

The master template mechanism was influenced by the Java Pet Store 1.0.1 example documented in the Java BluePrints. In this mechanism, one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel and providing a single source for updates. This mechanism is most useful for applications that can benefit from an explicitly controlled user interface template.

Appendix D, which discusses a "master template mechanism" in which "one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel" is clearly directed at Conallen's goal of "building model-driven Web applications." Appendix D and page 425 are directed at Web applications, not Web services. As noted above, Conallen is generally not directed at Web Services, but is instead generally directed at Web applications; Appendix D is clearly directed at Web applications. Furthermore, nowhere does Conallen extend the notions presented in his book regarding Web applications to generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components. Moreover, pages 64 and 425, cited by the Examiner, are from completely different and distinct sections of the Conallen reference directed at different technologies, and thus the Examiner is erroneously attempting to combine the disparate citations in a manner not described in the reference.

In further regard to claim 18, contrary to the Examiner’s assertion, Conallen does not anticipate means for generating one or more Use Cases for the Web Service in accordance with one or more design patterns. The Examiner cites Conallen, Fig. 6-11 on page 115: “develop use case model”; page 173: “use case to describe system behavior”; Fig. 8-5 on page 178: “browse catalog use case”; page 120: “design workflow”; pages 179-183: “modeling in UML”.

While the Conallen reference includes several sections that discuss various “use cases,” nowhere does Conallen disclose one or more Use Cases for a Web Service in accordance with one or more design patterns. As previously noted, Conallen is generally directed at building Web applications (“this book is about building model-driven Web applications”), not at building Web Services. The citations provided by the Examiner clearly disclose various Web application use cases. On page 173, one of the pages cited by the Examiner, at the beginning of a section titled “Use Cases”, Conallen actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications.

Furthermore, while Conallen does disclose various Web application-specific Use Cases, nowhere does Conallen disclose *means for generating one or more Use Cases for the Web Service in accordance with one or more design patterns*. In fact, Conallen does not even disclose *means for generating one or more Use Cases in accordance with one or more design patterns* in reference to the Web application-specific Use Cases Conallen does discuss.

In further regard to claim 18, contrary to the Examiner’s assertion, Conallen does not anticipate means for generating a high-level architecture for the Web Service. The Examiner cites Conallen, Fig. 8-4 on page 176: “top level use case diagram.” As previously noted, Conallen is generally directed at building Web applications, not at building Web Services. The citation provided by the Examiner clearly discloses a Web application use case. On page 173, at the beginning of a section

titled “Use Cases” that includes the Examiner’s citation, Conallen actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications

Applicant further notes that, on page 178, Conallen states (emphasis added):

The complete collection of use cases, actors, and diagrams form a use case model, which, like individual use cases, is just one part of the system’s requirement specification.

Thus, what Conallen illustrates on page 178 is clearly not even sufficient to be classified as a high-level architecture for a Web *application*, much less for a Web service.

In further regard to claim 18, contrary to the Examiner’s assertion, Conallen does not anticipate means for generating a high-level architecture for the Web Service in accordance with the one or more design patterns. The Examiner cites Fig. 8-7 on page 181: “browse catalog flow sequence.” Like Figure 8-4 cited above, Figure 8-7 on page 181 appears in the section titled “Use Cases” that begins on page 173. In this section, Conallen “concentrate[s] on the highlights and more interesting points [of use cases] as they relate specifically to Web-based applications.” As previously noted, Conallen is generally directed at building Web applications, not at building Web Services. In addition, on page 179, fourth paragraph, Conallen states:

Figure 8-7 shows a sequence diagram for the Browse Catalog use case basic flow scenario.

The citation provided by the Examiner clearly discloses a sequence pattern that may be used in developing a Web application use case. (“Another recommendation is to create a sequence diagram for each named use case scenario,” page 179, fourth paragraph, first sentence).

In further regard to claim 18, contrary to the Examiner’s assertion, Conallen does not anticipate wherein the high-level architecture identifies two or more entities of the Web Service. The Examiner cites Fig. D-3 on page 425: “main analysis

of class diagram in screen components”; page 438: “entity tier and data tier.” First, as previously noted, Conallen is generally directed at building Web applications, not at building Web Services. Figure D-3 on page 425 illustrates a class diagram for a Web application, and page 438 describes an Entity Tier and Data Tier for a Web application; neither citation teaches or suggests entities of a Web Service. Again, the Conallen reference itself clearly distinguishes between Web applications and Web services, and the citations provided by the Examiner clearly appear in portions of the Conallen reference that are directed at Web applications, not at Web services.

Furthermore, the Examiner previously cited Conallen, Fig. 8-4 on page 176: “top level use case diagram,” which appears in a section titled “Use Cases” that begins on page 173, as equivalent to Applicant’s “high-level architecture.” What Conallen describes on page 173 (the “top level use case diagram”) clearly does not identify what Conallen illustrates and describes on page 425 and page 438.

Furthermore, the Examiner cited Conallen, Fig. 8-4 on page 176: “top level use case diagram,” which appears in a section titled “Use Cases” that begins on page 173, and now cites Figure D-3 on page 425 which appears in Appendix D, titled “Master Template Pattern”, and page 438 which appears in a different section, Appendix E, titled “Glossary Application.” The Examiner has improperly cited different portions of Conallen directed at distinctly different aspects of Conallen’s teachings from different chapters and appendices of the Conallen reference and attempted to combine the citations to support the assertion that Conallen anticipates Applicant’s claim.

In further regard to claim 18, contrary to the Examiner’s assertion, Conallen does not anticipate wherein the high-level architecture identifies two or more entities of the Web Service and the relationships and interactions among the entities. The Examiner cites page 177: “relationship between use cases.” This citation appears in a distinctly different section of Conallen and appears to have little or nothing to do with the citations relied upon by the Examiner as teaching “entities” (Figure D-3 on page 425, and page 438). Neither Figure D-3 nor page 438 directly address or illustrate use cases.

Moreover, “use cases” are not themselves properly *entities*, and certainly would not be considered entities of a Web Service. Again, the Examiner has improperly cited different portions of Conallen directed at distinctly different aspects of Conallen’s teachings from different chapters and appendices of the Conallen reference and attempted to combine the citations to support the assertion that Conallen anticipates Applicant’s claim.

And again, page 177 appears in the section titled “Use Cases” that begins on page 173. In this section, Conallen “concentrate[s] on the highlights and more interesting points [of use cases] as they relate specifically to Web-based applications.” **The section is not directed at Web services at all.**

In further regard to claim 18, contrary to the Examiner’s assertion, Conallen does not anticipate means for generating a logical architecture for the Web Service according to the use case scenarios and in accordance with the one or more design patterns, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers. The Examiner cites page 237: “logical view of UML, server page, and client page”; Fig. 11-4 on page 241: “multiple forms in client pages”; Fig. 11-3 on page 239 “relationship among WAE elements”; Fig. 11-5 on page 241: “simple client page link association”; Fig. 11-6: “link associations originating from client page”; Table 11-1 on page 239: “HTTP, HTML”; page 240 and 242: “component view e.g. JSP, ASPX, ASCX, XML.” These citations all appear in a section titled “Web Application Extensions for UML” that begins on page 236. **The section is clearly directed at Web applications, and is not directed at Web services at all.** As previously noted, Conallen clearly distinguishes between Web applications and Web services, for example in the third paragraph on page 63:

The term Web Services is the latest hot phrase in development circles. Although the term has the word Web in it, it is not a Web application-specific technology.

The section in which the Examiner’s citations appear simply describes “the logical view of a UML model”, and clearly does not teach or suggest generating a logical

architecture for a Web service according to the use case scenarios and in accordance with one or more design patterns, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers, as is recited in claim 18.

Applicant reminds the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical invention must** be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Nowhere does the Conallen reference disclose “each and every element of the claimed invention” (claim 18 of the instant application) as arranged in the claim. Furthermore, even if Conallen did disclose one or more of the above elements, nowhere does Conallen disclose the above elements arranged as in claim 18. For at least the reasons given above, Conallen clearly does not anticipate Applicant’s claim 18.

Thus, for at least the reasons presented above, the rejection of claim 18 is not supported by the cited art and removal thereof is respectfully requested.

In regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate generating a vendor-independent Web Service architecture. As noted above, Conallen is generally not directed at Web Services, but is instead generally directed at Web applications. Furthermore, nowhere does Conallen extend the notions presented in his book to generating a vendor-independent Web Service architecture. The Examiner cites Conallen, page 65, and states “using UDDI, a standard for publishing and describing Web services” in support of the assertion that Conallen anticipates a system for generating a vendor-independent Web Service architecture. UDDI is **not** a system for generating a vendor-independent Web service architecture; UDDI is simply one component or tool that may be used in or with Web service architectures. As Conallen

points out on page 65, in the section cited by the Examiner, UDDI is instead “a mechanism for publishing and describing Web services to potential clients.” On page 65, Conallen further states “A UDDI Business Registry is a set of replicated registries of information about Web services on a network.” On page 66, Conallen states that the “general usage scenario [for UDDI] is for a programmer to use a Web-based interface or specialized tool to query the UDDI registry via its inquiry API.” UDDI is clearly not used to generate a vendor-independent Web Service architecture; UDDI is used to register and publish descriptive information about Web services on a network, and to query for the published descriptive information.

To further clarify that UDDI is not used to generate a vendor-independent Web Service Architecture, Conallen goes on to state, on page 66, “This information described in a UDDI business service...categorizes and points to URLs that describe Web services but doesn’t provide enough detail for a programmer to code a system that can accept and send SOAP-based Web service messages.”

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components. The Examiner cites page 425, “web server is most likely a commodity component, such as Tomcat, WebSphere, WebLogic and IIS.” Page 425 appears in Appendix D, titled “Master Template Pattern,” which begins on page 423, and presents a “Logical View.” The first paragraph on page 423, titled “Overview,” states (emphasis added):

The master template mechanism was influenced by the Java Pet Store 1.0.1 example documented in the Java BluePrints. In this mechanism, one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel and providing a single source for updates. This mechanism is most useful for applications that can benefit from an explicitly controlled user interface template.

Appendix D, which discusses a “master template mechanism” in which “one page template (JSP) is used for all outgoing pages, thereby helping enforce a consistent user interface look-and-feel” is clearly directed at Conallen’s goal of “building model-driven

Web applications.” Appendix D and page 425 are directed at Web applications, not Web services. As noted above, Conallen is generally not directed at Web Services, but is instead generally directed at Web applications; Appendix D is clearly directed at Web applications. Furthermore, nowhere does Conallen extend the notions presented in his book regarding Web applications to generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components. Moreover, pages 64 and 425, cited by the Examiner, are from completely different and distinct sections of the Conallen reference directed at different technologies, and thus the Examiner is erroneously attempting to combine the citations.

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components in accordance with one or more design patterns. The Examiner cites Conallen, Fig. 6-11 on page 115: “develop use case model”; page 173: “use case to describe system behavior”. Conallen does not disclose, in the citations or elsewhere, design patterns for use in generating a vendor-independent Web Service architecture comprising a plurality of heterogeneous components. The requirements set on page 115 that includes a “use case model” is a requirements set for a Web application, not for a Web service. Page 173, cited by the Examiner, actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate wherein said generating a vendor-independent Web Services architecture comprises generating one or more Use Cases for the Web Service. The Examiner cites Conallen, Fig. 8-5 on page 178: “browse catalog use case”; page 120: “design workflow”; pages 179-183: “modeling in UML”.

While the Conallen reference includes several sections that discuss various “use cases,” nowhere does Conallen disclose one or more Use Cases for a Web Service. As

previously noted, Conallen is generally directed at building Web applications (“this book is about building model-driven Web applications”), not at building Web Services. The citations provided by the Examiner clearly disclose various Web application use cases. On page 173, one of the pages cited by the Examiner, at the beginning of a section titled “Use Cases”, Conallen actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications.

Furthermore, while Conallen does disclose various Web application-specific Use Cases, nowhere does Conallen disclose *generating one or more Use Cases for a Web Service*.

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate wherein said generating a vendor-independent Web Services architecture comprises generating a high-level architecture for the Web Service. The Examiner cites Conallen, Fig. 8-4 on page 176: “top level use case diagram.” As previously noted, Conallen is generally directed at building Web applications, not at building Web Services. The citation provided by the Examiner clearly discloses a Web application use case. On page 173, at the beginning of a section titled “Use Cases” that includes the Examiner’s citation, Conallen actually states (emphasis added):

Because a full discussion of use cases is beyond the scope of this book, I will concentrate on the highlights and more interesting points as they relate specifically to Web-based applications

Applicant further notes that, on page 178, Conallen states (emphasis added):

The complete collection of use cases, actors, and diagrams form a use case model, which, like individual use cases, is just one part of the system’s requirement specification.

Thus, what Conallen illustrates on page 178 is clearly not even sufficient to be classified as a high-level architecture for a Web *application*, much less for a Web service.

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen

does not anticipate wherein the high-level architecture identifies two or more entities of the Web Service. The Examiner cites Fig. D-3 on page 425: “main analysis of class diagram in screen components”; page 438: “entity tier and data tier.” First, as previously noted, Conallen is generally directed at building Web applications, not at building Web Services. Figure D-3 on page 425 illustrates a class diagram for a Web application, and page 438 describes an Entity Tier and Data Tier for a Web application; neither citation teaches or suggests entities of a Web Service. Again, the Conallen reference itself clearly distinguishes between Web applications and Web services, and the citations provided by the Examiner clearly appear in portions of the Conallen reference that are directed at Web applications, not at Web services.

Furthermore, the Examiner previously cited Conallen, Fig. 8-4 on page 176: “top level use case diagram,” which appears in a section titled “Use Cases” that begins on page 173, as equivalent to Applicant’s “high-level architecture.” What Conallen describes on page 173 (the “top level use case diagram”) clearly does not identify what Conallen illustrates and describes on page 425 and page 438.

Furthermore, the Examiner cited Conallen, Fig. 8-4 on page 176: “top level use case diagram,” which appears in a section titled “Use Cases” that begins on page 173, and now cites Figure D-3 on page 425 which appears in Appendix D, titled “Master Template Pattern”, and page 438 which appears in a different section, Appendix E, titled “Glossary Application.” The Examiner has improperly cited different portions of Conallen directed at distinctly different aspects of Conallen’s teachings from different chapters and appendices of the Conallen reference and attempted to combine the citations to support the assertion that Conallen anticipates Applicant’s claim.

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate wherein the high-level architecture identifies two or more entities of the Web Service and the relationships and interactions among the entities. The Examiner cites page 177: “relationship between use cases.” This citation appears in a distinctly different section of Conallen and appears to have little or nothing to do with

the citations relied upon by the Examiner as teaching “entities” (Figure D-3 on page 425, and page 438). Neither Figure D-3 nor page 438 directly address or illustrate use cases. Moreover, “use cases” are not themselves properly *entities*, and certainly would not be considered entities of a Web Service. Again, the Examiner has improperly cited different portions of Conallen directed at distinctly different aspects of Conallen’s teachings from different chapters and appendices of the Conallen reference and attempted to combine the citations to support the assertion that Conallen anticipates Applicant’s claim.

And again, page 177 appears in the section titled “Use Cases” that begins on page 173. In this section, Conallen “concentrate[s] on the highlights and more interesting points [of use cases] as they relate specifically to Web-based applications.” **The section is not directed at Web services at all.**

In further regard to claim 20, contrary to the Examiner’s assertion, Conallen does not anticipate wherein said generating a vendor-independent Web Services architecture comprises generating a logical architecture for the Web Service according to the use case scenarios, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers. The Examiner cites page 237: “logical view of UML, server page, and client page”; Fig. 11-4 on page 241: “multiple forms in client pages”; Fig. 11-3 on page 239 “relationship among WAE elements”; Fig. 11-5 on page 241: “simple client page link association”; Fig. 11-6: “link associations originating from client page”; Table 11-1 on page 239: “HTTP, HTML”; page 240 and 242: “component view e.g. JSP, ASPX, ASCX, XML.” These citations all appear in a section titled “Web Application Extensions for UML” that begins on page 236. **The section is clearly directed at Web applications, and is not directed at Web services at all.** As previously noted, Conallen clearly distinguishes between Web applications and Web services, for example in the third paragraph on page 63:

The term Web Services is the latest hot phrase in development circles. Although the term has the word Web in it, it is not a Web application-specific technology.

The section in which the Examiner's citations appear simply describes "the logical view of a UML model", and clearly does not teach or suggest *generating a logical architecture for a Web service according to the use case scenarios and in accordance with one or more design patterns, wherein the logical architecture identifies two or more logical components of the Web Service and the relationship among the logical components, and wherein the logical architecture comprises two or more layers*, as is recited in claim 20.

In further regard to claim 20, contrary to the Examiner's assertion, Conallen does not anticipate implementing the Web Service according to the Web Service architecture. The Examiner cites pages 9-10 and Fig. 2-1: "build Web application based a basic web system on a Web Server." Applicant cannot identify the quote from the Examiner on the cited pages. However, Applicant notes that Fig. 2-1 illustrates a "Basic Web system," and is not a Web service. Furthermore, the discussion on pages 9-10 is directed at Web applications. Again, Conallen himself clearly distinguishes between Web applications and Web services. Conallen makes clear the distinction between Web Services and Web applications, for example in the third paragraph on page 63:

The term Web Services is the latest hot phrase in development circles. Although the term has the word Web in it, it is not a Web application-specific technology.

Conallen does not anticipate a method for generating a Web service architecture at all, and Conallen clearly does not anticipate implementing a Web Service according to a generated Web Service architecture.

Applicant reminds the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Nowhere does the Conallen reference

disclose “each and every element of the claimed invention” (claim 20 of the instant application) as arranged in the claim. Furthermore, even if Conallen did disclose one or more of the above elements, nowhere does Conallen disclose the above elements arranged as in claim 18. For at least the reasons given above, Conallen clearly does not anticipate Applicant’s claim 20.

Thus, for at least the reasons presented above, the rejection of claim 20 is not supported by the cited art and removal thereof is respectfully requested.

In regard to claim 38, claim 38 recites a computer-accessible storage medium including program instructions that are computer-executable to implement the method described above regarding claim 20. The Examiner’s rejection of claim 38 is not substantially different than the rejection of claim 20. Therefore, Applicant traverses the Examiner’s rejection of claim 38 for at least the reasons presented above in regards to claim 20.

Thus, for at least the reasons presented above, the rejection of claim 38 is not supported by the cited art and removal thereof is respectfully requested.

Applicant also asserts that the rejection of numerous ones of the dependent claims is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicant submits the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-66300/RCK.

Respectfully submitted,

/Robert C. Kowert/
Robert C. Kowert, Reg. #39,255
Attorney for Applicant

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: May 5, 2008